Zurich Research Laboratory

Improving Efficiency and Enhancing Concurrency of Untrusted Storage

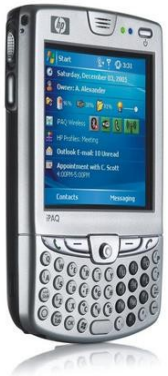
Christian Cachin <cca@zurich.ibm.com>
Idit Keidar <idish@ee.technion.ac.il>
Alexander Shraer <shralex@tx.technion.ac.il>

FAST 2008
27 February 2008

Where is my data?

1980

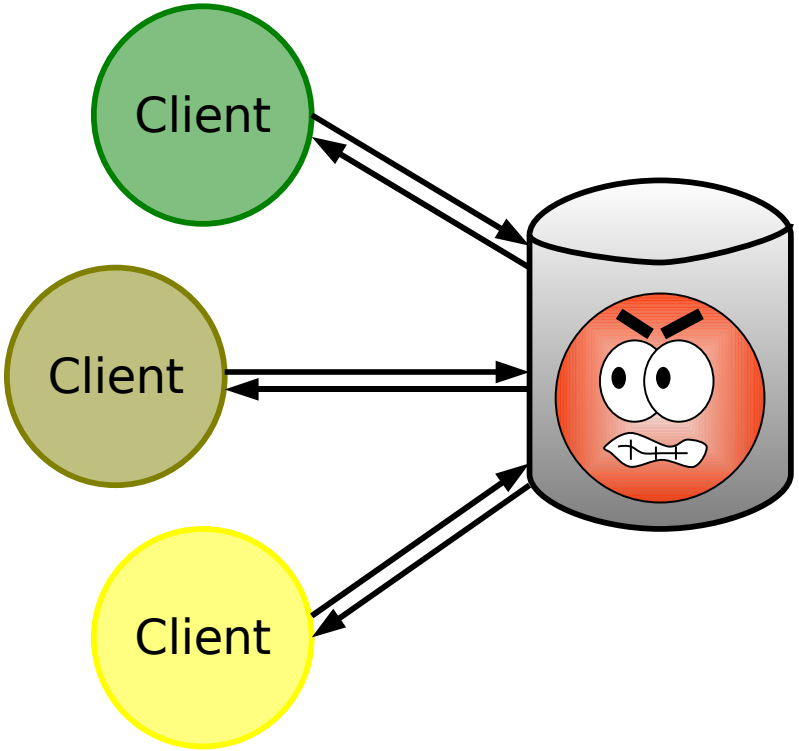
Google



IBM

2008

Untrusted Storage Service



- Many independent clients
  - Correct
  - Store data on server
  - Communicate only with server
  - Small trusted memory
- Storage server
  - Untrusted
  - Potentially corrupted
- Clients read and write concurrently

How to ensure consistent view of data to all clients?

Consistent Access to Untrusted Storage

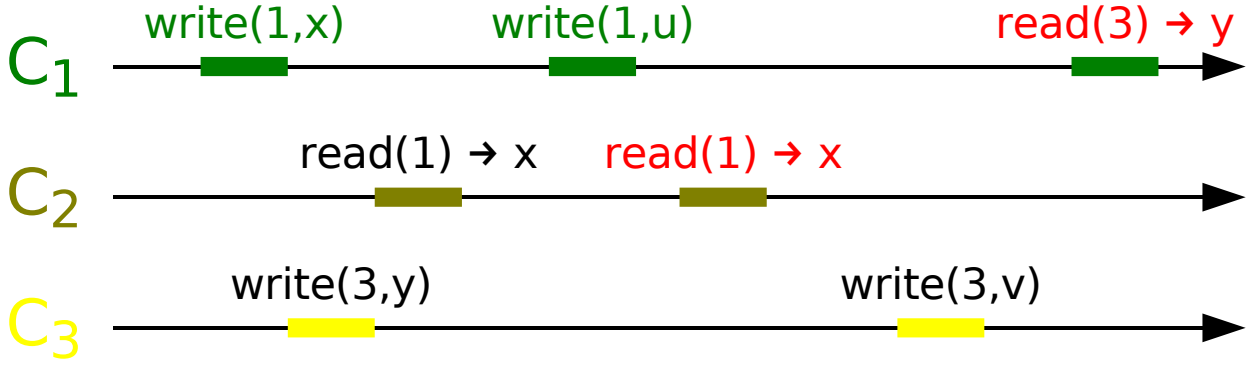
- Loose synchronization and concurrency pose a new problem
- Suppose clients sign data with digital signatures:
  Server cannot forge any values ...
  - → But answer with outdated value (“replay attack”)
  - → Or send different values to different clients
- Server may present different views to clients
  - → “Fork” their views of history
  - → Clients cannot prevent this
- Fork linearizability [MS02], provided by SUNDR [LKMS04]

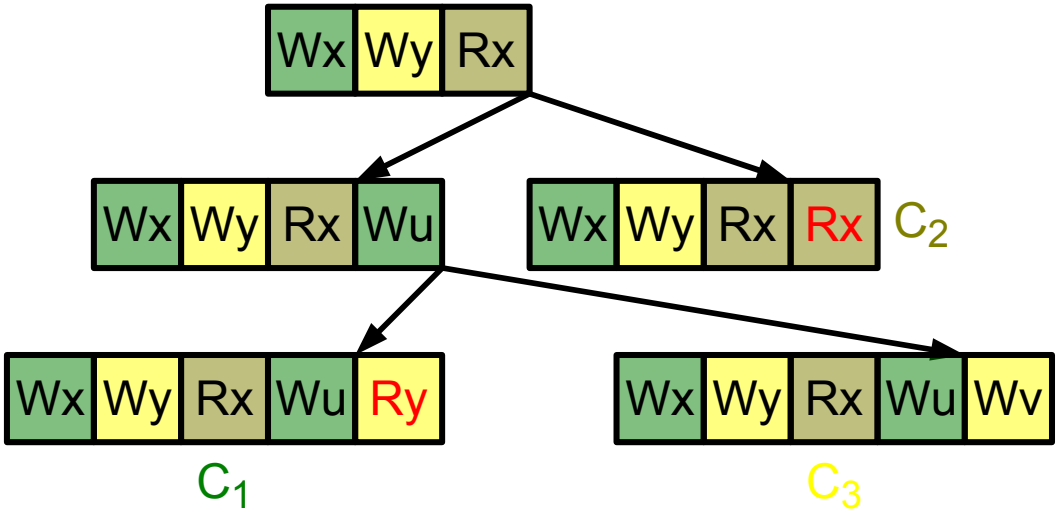
  If server forks the views of two clients *once*, **then**

  → their views are forked *ever after*

  → they *never again* see any updates of each other
- Forks are easier to detect than subtle data modifications
  - → Using a separate channel for detection

Fork-linearizability





After C_1 writes u,

C_2 reads x:

→ C_2 forked from C_1C_3

After C_1 reads y:

→ C_1 forked from C_3

New Results

- More efficient fork-linearizable communication protocol [CSS07]
  - → Messages of size $O(n)$ instead of $O(n^2)$ with n clients
- Fork-linearizable protocols are not wait-free [CSS07]
  - → Reader must wait for writer even if server correct
- New notion: weak fork-linearizability [CKS08]
  - → New wait-free protocol, where clients need not wait for each other and messages of size $O(n)$ only
- More impossibility results [CKS08]
  - → Fork-sequential consistency does not enable wait-free protocols
  - → Fork-* consistency does not enable wait-free protocols

References

- [CSS07] C. Cachin, A. Shelat, and A. Shraer. Efficient fork-linearizable access to untrusted shared memory. In Proc. 26th ACM Symp. Principles of Distributed Computing (PODC), 2007.
- [CKS08] C. Cachin, I. Keidar, and A. Shraer. Wait-free untrusted storage. Manuscript, Feb. 2008.
- [LKMS04] J. Li, M. Krohn, D. Mazières, and D. Shasha. Secure untrusted data repository (SUNDR). In Proc. Symp. Operating Systems Design and Implementation (OSDI), 2004.
- [MS02] D. Mazières and D. Shasha. Building secure file systems out of Byzantine storage. In Proc. 21st ACM Symp. Principles of Distributed Computing (PODC), 2002.